

Tópicos Especiales En Telemática

Reto 3: Aplicación distribuida tipo Publicador / subscriber utilizando un middleware orientado a mensajes

Presentado por:

John Jairo Tangarife Vélez

Juan Pablo Pineda

Docente:

Edwin Montoya

Universidad EAFIT Mayo del 2013

Contenidos

| | |
|--|---|
| 1Análisis de la aplicación..... | 1 |
| 1.1Descripción de la aplicación..... | 1 |
| 1.2Consideraciones iniciales..... | 1 |
| 2Diseño Global del sistema y arquitectura..... | 2 |
| 2.1Arquitectura seleccionada..... | 2 |
| 2.2Diseño Global del sistema..... | 2 |
| 2.3Descripción de componentes..... | 2 |
| 3Especificaciones de diseño y capas..... | 3 |
| 3.1Arquitectura Detallada..... | 3 |
| 3.2Capas del sistema..... | 3 |
| 3.3Descripción de las entidades..... | 4 |

1 Análisis de la aplicación

1.1 Descripción de la aplicación

Sistema de anuncios que permite a un conjunto de clientes obtener mensajes de anuncios de productos o servicios. El sistema tiene dos módulos principales: Proveedor de anuncios (AdFuente), es decir, es desde donde se generan los mensajes y un Cliente de anuncios (AdCliente), el cual recibe de manera asincrónica mensajes enviados a un Canal por un AdFuente.

1.2 Consideraciones iniciales

- El sistema debe tener un sistema de gestión que permita hacer canales a través de los cuales fluyen los Mensajes originados en un AdFuente (pueden haber varias fuentes en un mismo Canal) hacia uno o más AdCliente.
- Los mensajes que son enviados por un AdFuente hacia un Canal y que le debe llegar a los AdCliente, estén disponibles para que los reciba el AdCliente los reciba al momento de conectarse.
- El AdCliente recibe los mensajes inmediatamente son publicados si se

encuentra conectado.

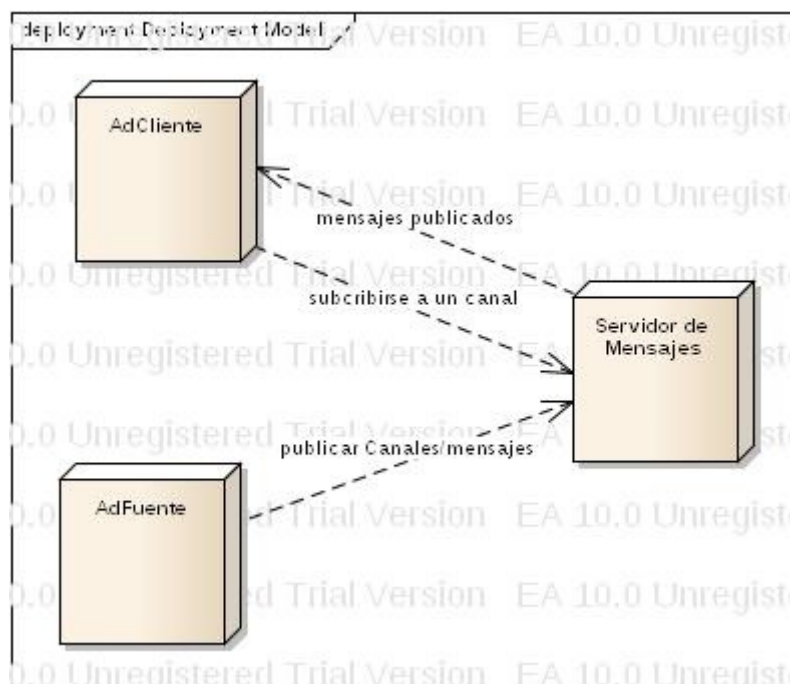
- Los Canales son temáticos (deportes, tecnología, noticias, culinaria, bolsa, etc) y deben ser gestionados en el sistema (crear, modificar, borrar, etc un canal) por el AdFuente.
- Los AdCliente podrán recuperar los mensajes que no halla recibido de un canal determinado o permanecer en modo “escucha” para ir recibiendo mensajes a medida que se producen.
- A medida que los AdCliente suscritos a un Canal consumen los mensajes, estos se irán descolando.
- Se pueden crear mensajes en un canal con vigencia de tiempo determinada. Es decir, pasado un tiempo determinado, los AdCliente no pueden recibir estos mensajes.
- EL AdCliente puede suscribirse y retirarse de un canal disponible en el servidor de mensajes a voluntad.

2 Diseño Global del sistema y arquitectura

2.1 Arquitectura seleccionada

El sistema tendrá una arquitectura cliente/servidor, con dos tipos de clientes: AdCliente que recibe los mensajes y se suscribe y retira de los canales; y AdFuente que es un cliente que crea y gestiona canales y publica mensajes. El servidor será un servidor apache-activeMQ.

2.2 Diseño Global del sistema



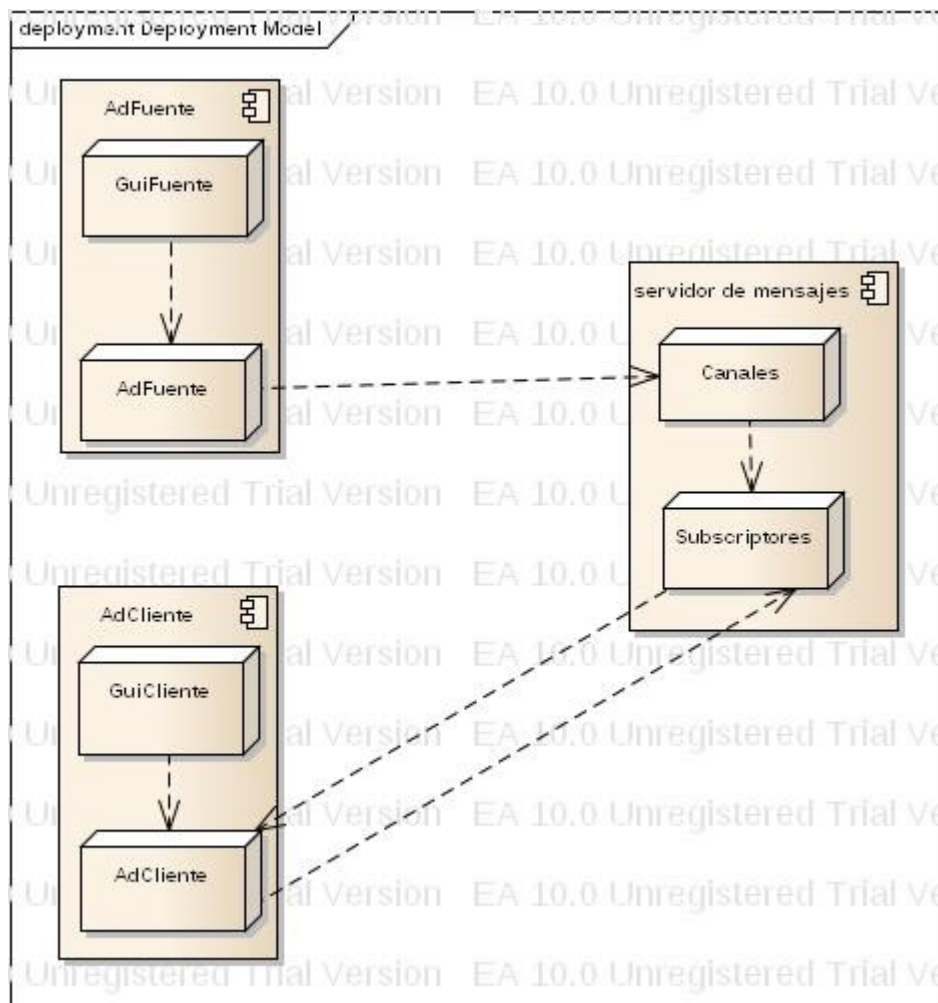
2.3 Descripción de componentes

- AdCliente: componente cliente del diseño que recibe de manera asincrónica los mensajes publicados en el servidor de mensajes y además se suscribe a canales específicos para recibir estos mensajes.

- AdFuente: componente cliente del diseño que crea los canales para publicar mensajes y además publica los mensajes en esos canales.
- Servidor de mensajes: es el lugar donde se almacenan los canales con sus mensajes y que permite la comunicación entre los dos tipos de clientes.

3 Especificaciones de diseño y capas

3.1 Arquitectura Detallada



3.2 Capas del sistema

La arquitectura por capas se implementó en los componentes cliente AdCliente y AdFuente.

Capa de Presentación: Esta conformada por las entidades GuiCliente y GuiFuente. Le presenta a un usuario un menú con las opciones de los servicios que presta la aplicación.

Capa Lógica: Está conformada por las entidades AdFuente y AdCliente que contienen los métodos que permiten el manejo de canales y envío y recepción de mensajes.

3.3 Descripción de las entidades

- **GuiFuente:** Entidad que ofrece un menú a los usuario para que puedan gestionar Canales y enviar mensajes a los canales.
- **GuiCliente:** Entidad que ofrece un menú a los usuarios para que puedan suscribirse a canales, recibir mensajes de manera asincrónica de los canales a los que están suscritos y eliminar suscripciones.
- **AdCliente:** entidad que contiene la lógica que permite suscribirse a canales y traer mensajes de estos canales por medio de los siguientes métodos:
 - **onMessage(mensaje):** método de la interfaz `MessageListener` que se sobrescribe para que reciba los mensajes de determinado canal a medida que estos se van publicando.
 - **subscribirse(nombreCanal):** método que permite realizar una suscripción durable a un canal.
 - **dejarCanal(subscriptor):** método que elimina una suscripción a un canal específico.
 - **TraerCanales():** método que permite traer los canales disponibles en el servidor de mensajes.
 - **modoRecibir(nombreCanal, modo):** método que permite traer los mensajes de un canal específico y permite dos modos: solo traer los mensajes que no ha recibido y traer los mensajes que no ha recibido y quedarse recibiendo los mensajes que sean publicados.
 - **setCedula(cedula):** método que permite almacenar la identidad del usuario para que pueda crear y eliminar suscripciones.
- **AdFuente:** entidad que contiene la lógica que permite gestionar canales y publicar mensajes en estos canales por medio de los siguientes métodos:
 - **crearCanal(nombreCanal):** método que permite crear un canal en el servidor de mensajes.
 - **publicar(mensaje):** método que permite publicar mensajes en un canal existente.
 - **verificarCanal(canal):** método que permite verificar la existencia de un canal.
 - **TraerCanales():** método que permite ver los canales existentes en el servidor de mensajes.
- **Canales:** entidad que es un tópico que permite almacenar mensajes que son enviados a sus suscriptores de manera asincrónica.
- **Subscriptores:** entidad creada con un identificador único que relaciona a un usuario con un canal específico.